

Computer Systems Project Proposal

Graphical Display of Physics Simulation (Java)

Steven Oetjen

November 2, 2006

1 Problem Statement

A physics simulation, in order to adequately demonstrate physical laws and predict an unlimited number of scenarios, must implement a broad range of mathematical equations and provide the user with the ability to set up a scenario with whatever number of objects and arrangements of these objects that he desires. The goal of this project is to create such a simulation.

2 Purpose

The purpose is to make a physics simulation that will display objects placed by the user graphically and display information about those objects on a graphical user interface. The goal is to allow the user to place any combination of objects, including particles, springs, and ramps, in a graphical display, input values for these objects, such as constants, coefficients, and variables, and run a simulation that will track these values and display the interactions and positions of the objects graphically in two-dimensions. This is worthwhile because it provides an opportunity to quickly and easily test or verify phenomena. It is a good project for the Computer Systems Lab because it makes use of a computer's ability to store, manipulate, and display data to make a simulation. The results may attract the interest of many, including students needing an aid in their understanding of the physical world and how it behaves. The results can also be applied to making or confirming predictions on how certain situations will be resolved.

3 Scope of Study

Bounds will be limited to two-dimensions with particles, springs, and ramps, thus constricting the field of study. The concepts involved are kinematics, dynamics and Newton's laws, energy and work, conservation of momentum and collisions, gravitational force, and electric charge, field, potential, and force. Variables such as mass, displacement, kinetic energy, power, and charge, only to name a few, are required, as well as the relationships between these variables in the form of equations.

4 Background Research

Basic information about theoretical physics, both conceptual explanations and (non-calculus) equations, are provided by Douglas C. Giancoli's text, *Physics: Principles with Applications*. A supplementary website for the textbook is:

http://wps.prenhall.com/esm_giancoli_physicsppa_6/0,8713,1113739-,00.html

Maciej Matyka's "How To Implement a Pressure Soft Body Model" explains his method of implementing this model, and outlines his algorithm, which takes into account gravitational and spring forces, as well as pressure. It also implements a first order Euler integrater to calulate force accumulations. The paper shows the mathematical analyses behind the calculation algorithms for gravity force, spring linear force, volume of the body, and pressure force distribution. The results of his example 2d softball model are shown, as well as a three dimensional version of the pressure soft body model. This article can be found at:

<http://panoramix.ift.uni.wroc.pl/~maq/soft2d/howtosoftbody.pdf>

A simulation very similar in quality to the one I wish to create can be found at the "My Physics Lab" page. This has a series of java applets that run various scenarios based on user input for chosen quantities. These scenarios involve linear springs, pendulums, colliding blocks, roller coasters, and molecules. It also can graph the change in certain quantities over time. This page can be found at:

<http://www.myphysicslab.com/>

My approach varies from Matyka's soft body model in that my simulation will consist of several separate rigid bodies, and so will not need to go into the complexity of pressure force distribution. It will, however, use linear spring force and gravitation between the rigid particles, and also will possibly use some sort of integration method to improve the accuracy of the calculations. The result will be much more similar to "My Physics Lab" in that the scope over physics topics will be much broader, and separate bodies of different types and quantities will be used. The major difference is that my simulation will not consist of a choice of preconstructed scenarios, but the user will be able to add any number of objects and orient them however he wishes, so there is no limit to the scenarios that can be tested.

5 Procedure and Methodology

The project will be divided into several iterations of design, programming, and testing. The tasks of these iterations include implementing each concept and set of equations and laws. These groups are kinematics, dynamics, momentum, gravitation, energy, and electrostatics. Each iteration will proceed with the following steps:

Design Classes, methods, and variables will be added or modified, and the function of each element will be determined. The means of communication of information between relevant classes will be figured.

Programming The necessary changes to the program will be made. All programming will be done in Java, using Swing and AWT packages for graphics and GUI components.

Testing To test a given implementation of a law, the user will input arbitrary values for all necessary fields involved in or dependent on that law. The output generated by these random types of input will be compared to theoretical values derived from mathematical formulas. To analyze the error, a percent error calculation will be performed on these two values, and the accuracy will be gauged by that percent. In this way, the accuracy of the program's ability to model predictable phenomena can be measured and displayed in chart format.

Aside from implementing equations and laws, several other components to the program must be implemented.

1. GUI Structure and Layout
2. Graphics Panel Distance and Time Scale
3. Support of Multiple Objects/Multiple Types and Classes of Objects
4. Support of Coordinate and Polar Vector Conversions
5. Collision Detection
6. Improved Integration Methods

The first two of these tasks should be completed first as a basic set up of the program's input and output structure. The order in which the remaining tasks are completed should fit logically with the addition of physics laws. For example, the implementation of kinematics equations can be done without first supporting multiple objects, because all that is needed is one object that stores position, velocity, and acceleration. After allowing support of multiple objects, the next two tasks could be implementing conservation of momentum and detecting collisions. This is logical because the law of conservation of momentum is needed to properly determine the outcomes of collisions. Thus, complementary tasks should be completed together to help test each other, and other tasks in general can be completed in order of need or increasing complexity.

6 Expected Results

The expected result of this project is a helpful and accurate representation of the behavior of objects in the physical world, with reasonable speed, relative to real time. This could be used to demonstrate and experiment certain phenomena to others, aid in students' understanding, and even predict the outcomes of certain scenarios.

The results and analyses can be presented by modeling existing or expected phenomena, with inputs that are easy for others to verify correctness. This is preferred over using arbitrary inputs, such as the ones used in testing. Rather than using random meaningless

values, clear scenarios that demonstrate common physical occurrences will be used. This will increase interest in the project, as well as emphasize its predictive and demonstrative capability. These results will give future researchers verification or confirmation of results for a given scenario, and may even aid in visual or practical understanding of a scenario.